逻辑与神经之间的桥

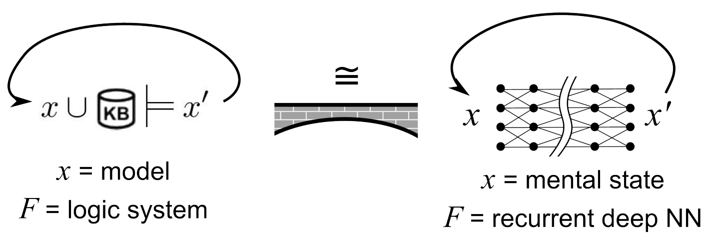
甄景贤 (King-Yin Yan)

General.Intelligence@Gmail.com

Abstract. Logic-based AI 和 connectionist AI 长久分裂，但笔者最近发现了可以统一两者的理论。

逻辑 AI 那边，「结构」很精细，但学习算法太慢；我的目的是建立一道「桥」，将逻辑 AI 的某部分结构**转移**到神经网络那边，这样可以融合两边的好处。

这个问题搞了很久都未能解决，因为逻辑 AI 那边的结构不是一般常见的数学结构，单是要要表述出来也有很大困难。直到我应用了 model theory 的观点，才找到满意的解决方法：



(1)

首先解释 logic 那边的结构，然后再解释 neural network 那边的结构。

1 逻辑的结构

一个逻辑系统可以这样定义：

- 一些 constant symbols，predicate symbols，和 function symbols
- 由上述的原子建立 命题 (propositions)
- 命题之间可以有连接词：$\neg, \wedge, \vee$ 等
- 建立 逻辑后果 (consequence) 关系：$\Gamma \vdash \Delta$

我个人认为 relation algebra [7] [5] 比较接近人类自然语言，但在数理逻辑研究中最通用的逻辑是 first-order logic (FOL)。然而这并不是重点，因为各种逻辑基本上是等效的，而且相互之间可以很容易地转换。以下集中讨论 FOL。

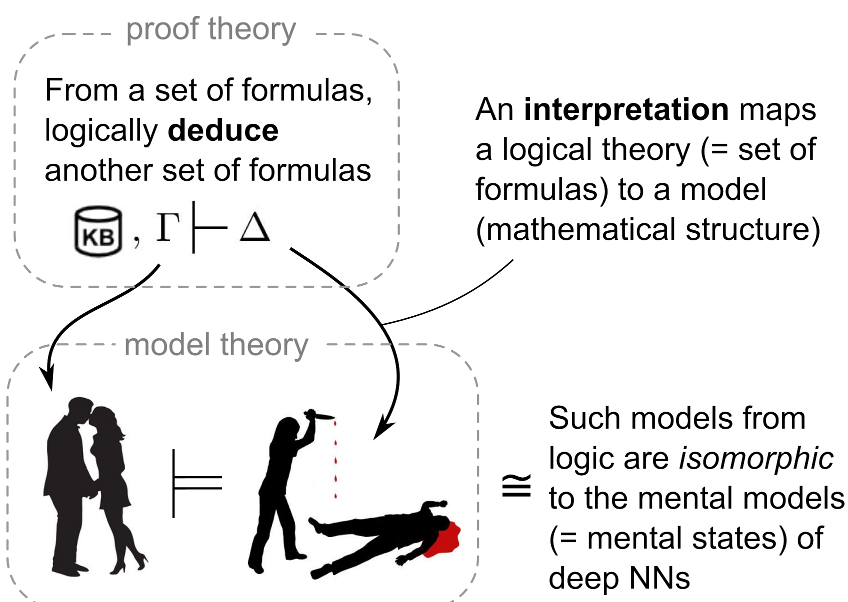
我以前花了很多时间思考怎样将逻辑的 $\vdash$ 关系过渡到神经网络去，但发觉这个目标非常 elusive。

一方面，逻辑是几百年来发展起来的关於人类思考的规律；逻辑的描述是正确的；逻辑和神经之间必然有一个 correspondence，因为它们都在做同样的事：智能。

在认知科学里，有很多人相信大脑的内部的 representation 是一些所谓 “mental models”，而很少人会相信大脑使用一些像命题那样的符号结构做 representation，甚至用 λ-calculus 那样的符号 manipulation 去思考。

举例来说，用文字描述一起凶杀案，读者心目中会建立一个「模型」，它类似於真实经验但又不是真实的。人脑似乎是用这样的 mental models 思考，而不是一些命题的集合。

所以我终於发现到，logic-neuro correspondence 必须透过 model theory [3] [6] 才能达成：



(2)

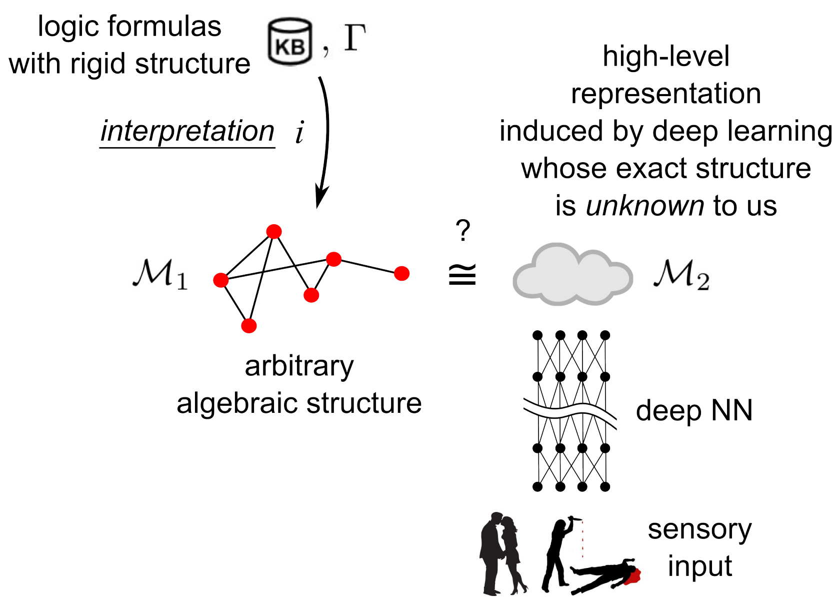
2 Model theory

如果用範畴论的方法表示：

$$\begin{tikzcd} \mathcal{L} \arrow[d, "i"] & \\ \mathcal{M}_1 \arrow[r, phantom, "\simeq"] & \mathcal{M}_2 \\ & \mathcal{S} \arrow[u, "\text{dNN}"'] \end{tikzcd} \tag{3}$$

- $\mathcal{L}$ = category of logic theories (= sets of formulas)
- i = interpretation maps
- $\mathcal{M}_1$ = category of models (from logic)
- $\mathcal{M}_2$ = category of models (from deep NNs)
- $\mathcal{S}$ = sensory input

上图等同於下面的卡通解释：



(4)

换句话说，$\mathcal{M}_2$ =  是由深度学习 induce 出来的结构；但它的结构对我们来说是不透明的（这是神经网络的弱点）。

而 $\mathcal{M}_1 =$ 的结构是 free 的；换句话说，那 i map 的 source domain 是固定的，但 target domain 是自由的。这导致 i map 的学习很困难，因为 $\mathcal{M}_1$ 和 $\mathcal{M}_2$ 的结构都不清楚。必须更详细分析 $\mathcal{M}_1, \mathcal{M}_2$ 的结构。

3 Model 和 interpretation 的结构

在模型论中，$\mathcal{L}$ 是逻辑句子的範畴，$\mathcal{M}_1 =$ 可以是任何抽象代数结构。只需把 $\mathcal{L}$ 中的 constants, predicates, relations, functions 映射到 $\mathcal{M}_1$ 就行。为简化讨论，我们只考虑 constants 和 relations，因为二者是逻辑中最**本质**的东西。

$$\begin{array}{rcl} \mathcal{L} & \xrightarrow{i} & \mathcal{M}_1 \\ \text{constant symbol} & \overset{i}{\mapsto} & \bullet \\ \text{relation symbol} & \overset{i}{\mapsto} & \bullet\!\!-\!\!\bullet \end{array} \tag{5}$$

问题是在神经那边缺乏 的结构。一直以来，人们习惯把神经网络看成是 "black box"，但如果我们不知道 $\mathcal{M}_2 =$ 的结构，就无法建立 $\mathcal{M}_1 \simeq \mathcal{M}_2$ 的 isomorphism。

4 神经网络的结构

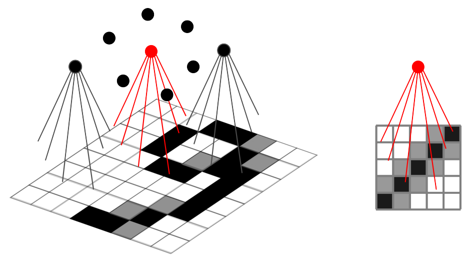
那么，神经网络的 representation 究竟是什么结构？

一个**神经网络**基本上是：

$$F(\boldsymbol{x}) = \sigma(W_1 \sigma(W_2 ... \sigma(W_L\ \boldsymbol{x}))) \tag{6}$$

其中 L 是层数，W 是每层的权重**矩阵**，σ 是对每个分量的 sigmoid function（其作用是赋予非线性）。

考虑最简单的情况，例如提取 digit "9" 的特徵的一层网络。这层网络可以有很多神经元（左图），每个神经元局部地覆盖输入层，即所谓视觉神经元的 local receptive field（右图）。



(7)

假设红色的神经元专门负责辨识「对角线」这一特徵。它的方程式是 $\boldsymbol{y} = (W\boldsymbol{x})$。矩阵 W 的作用是 affine「旋转」特徵空间，令我们想要的特徵指向某一方向。然后再用 「**挤压**」想要的特徵和不想要的特徵[1]。Sigmoid 之后的输出，代表某类特徵的**存在与否**。

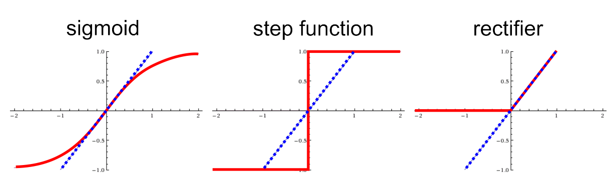
换句话说：每个神经元的输出其实代表某个 feature 的存在与否。
而，更高层的神经元代表下层 features 之间的**关系**。

凭这个思路推广，可以推测这样的 correspondence：

$$\begin{array}{rcl} \mathcal{M}_1 & \simeq & \mathcal{M}_2 \\ \text{constant} & \Leftrightarrow & \text{neuron} \\ \text{relation} & \Leftrightarrow & \text{relation between higher and lower neurons} \end{array} \tag{9}$$

但要注意的是这对应未必是一对一的，可能是一个 constant 对应几个 neurons 的**线性组合**。具体情况可能像以下的示意图（实际上每层神经网络可能有很多

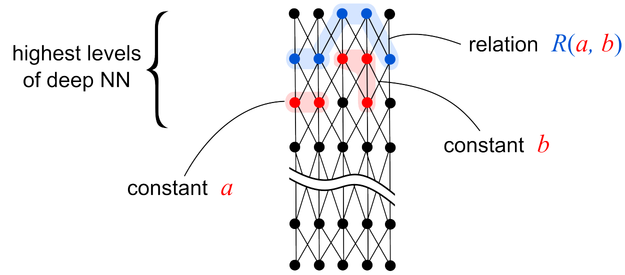
[1] $^{-1}$ 的作用是「扯」(stretch)，将本来邻近的两点的距离非线性地拉远。看看以下各种常见的激活函数，它们全都是相对於 identity $y = x$ 的非线性 deformation：



(8)

这和 Steven Smale 提出的「马蹄」[8] 非常类似，它是制造混沌的处方之一。换句话说，「拉扯」然后放回原空间，如此不断重复，就会产生混沌 [4] [9]。其作用类似於「搓面粉」，所以另一个变种也叫做 baker map。

神经元）：



(10)

$R(a,b)$ 可以在 a,b 的 common parents 中寻找（例如那些蓝色神经元，$R(a,b)$ 的值 = 蓝色神经元的某个线性组合）。验证的方法是：当 a 和 b 的信号都是「有」时，$R(a,b)$ 的值也应该是 true。

看上去颇复杂，但这样已经可以直接由逻辑式子 $\mathcal{L}$ 映射到深度网络的输出层。在未有这理论之前，完全不知道这个 map 的结构；但现在假如理论是正确的话，只需要简单的组合搜索 (combinatorial search) 就可以找到对应。举例来说，对於用深度学习做 natural language understanding 的人，这理论或许会很有用。

5 Prior art

Bader, Hitzler, Hölldobler and Witzel 在 2007 年提出了一个 neural-symbolic integration 的做法 [2]。他们首先由 logic theory 生成抽象的 Herbrand model，再将 Herbrand model 映射到某个 fractal 空间，然后直接用神经网络学习那空间。虽然用了 model theory，但他们没有利用到本文所说的 $\mathcal{M}_1$ 和 $\mathcal{M}_2$ 之间的关系。

Acknowledgement

谢谢 Ben Goertzel（OpenCog 人工智能的创始人）在 AGI mailing list 上和我的讨论。Ben 初次指出神经网络学习和逻辑 inductive 学习的不同，引起我研究两者之间的关系。

References

1. Itamar Arel. *Deep reinforcement learning as Foundations for Artificial Intelligence*, chapter 6, pages 89–102. Atlantis Press, 2012.
2. Bader, Hitzler, Hödobler, and Witzel. The core method: Connectionist model generation for first-order logic programs. *Studies in Computational Intelligence 77, 205-232*, 2007.

3. Kees Doets. *Basic model theory.* CSLI notes, 1996.
4. Robert Gilmore and Marc Lefranc. *The topology of chaos: Alice in stretch and squeezeland.* Wiley-VCH, 2011.
5. Roger Maddux. *Relation algebras.* Elsevier, 2006.
6. Maria Manzano. *Model theory.* Oxford, 1999.
7. Gunther Schmidt. *Relational mathematics.* Cambridge, 2010.
8. Stephen Smale. Differentiable dynamical systems. *Bulletin of the American Mathematical Society*, 1967.
9. Tamás Tél and Márton Gruiz. *Chaotic dynamics: an Introduction based on classical mechanics.* Cambridge, 2006.
10. King Yin Yan. Wandering in the labyrinth of thinking – a cognitive architecture combining reinforcement learning and deep learning. to be submitted AGI 2017.